REMARKS

Claims 1-17 remain pending in the application.

Claims 1-17 over Eaton and Feemster

In the Office Action, claims 1-17 were rejected under 35 U.S.C. §103(a) as allegedly being obvious over U.S. Pat. No. 3,924,245 to Eaton et al. ("Eaton") in view of U.S. Pat. No. 5,608,873 to Feemster et al. ("Feemster"). The Applicants respectfully traverse the rejection.

Claims 1-7 recite a shared memory accessible by a first processor and a second processor, and including a first mailbox portion to **pass data** from the first processor to the second processor. Claims 8-17 recite first direction messages **passed** from a first processor to a second processor, and second direction messages **passed** from the second processor to the first processor.

Eaton is old art teaching the use of a stack mechanism for a data processor. The Examiner cites Eaton for allegedly teaching a "shared memory including a first mailbox portion to pass data from said first processor to said second processor, and a second mailbox portion to pass data from said second processor to said first processor". (Office Action at 2) The Examiner cites the passage at col. 4, lines 7-22 in support of this allegation.

Eaton teaches the use of memory stacks, which were techniques to push data for use by a single processor in a First In, Last Out approach. In the cited section of col. 4, Eaton teaches the use of stacks for a single processor that grow towards one another. This disclosure relates to the use of memory by a single processor, not to **shared** memory as claimed by claims 1-17, much less shared memory used to **pass data** from a first processor to a second processor as claimed by claims 1-17.

In the very last paragraph of Eaton's disclosure, at col. 6, lines 42-53, Eaton makes the very first and only disclosure relating to multiple processors, with the following:

> In another modification of the system described above, the system includes two separate processing units which share the same microprogram store 11, each unit being allocated a separate area of the microprogram store for containing its microprogram. The

> units also share the main store 10. In this case, the overlay table 17 is extended, so that each entry now contains one set of fields VL, VA, VSA for an overlay relating to one of the processing units, and a similar set of fields for an overlay relating to the other unit. In addition, two sets of registers 27 must now be provided, one for each of the processing units. (emphasis added)

Thus, Eaton makes it imminently clear that the processors are making entirely separate use of the microprogram store 11. There is no passage of data from one processing unit to the other as claimed in claims 1-17.

The Examiner makes a huge leap in assuming that the **stacks** disclosed by Eaton are 'mailboxes'. There is absolutely no support for the notion of passing data between processors whatsoever in Eaton, much less the use of mailboxes in shared memory to accomplish such. Eatons STACKS are NOT MAILBOXes. They contain data which fetched program instructions execute upon. (Eaton, col. 2, lines 21-33)

The Examiner's assumption in the very first sentence of the rejection on page 2 of the Office Action, that "Eaton teaches a shared memory processor-to-processor mailbox between at least two processors" from the single paragraph at the end of the disclosure is entirely misplaced. Eaton does NOT teach processor-to-processor communications AT ALL, much less a processor-to-processor mailbox as alleged. Eaton at best teaches a technique for placing a stack from one processor in a same memory store as a stack for another processor.

The Examiner agrees that Eaton fails to teach a first processor having write access to a first mailbox portion and not to the second mailbox portion (Office Action at 3), but alleges that Feemster's **mailbox** teachings would be applicable to Eaton's processor **stack** used for data upon which computer instructions would operate. Feemster's mailbox would be quite useless as a data stack for instructions to operate upon, as Eaton discloses.

Eaton and Feemster are entirely different uses of shared memory. Eaton teaches the use of a common memory for multiple data stack usage by the **same** processor. The single paragraph at the end of the disclosure which mentions the use of two processors merely extends the thought that data stacks

for another processor may also be defined in the common memory store. However, Eaton includes absolutely NO TEACHING of PASSING DATA between processors.

The Examiner appears to believe that it is a straight forward, simple task to pass data between processors in a shared memory. It is NOT. Separate processors operate on separate clock cycles, separate speeds, etc., and collisions are inevitable over time if the same memory location is accessed simultaneously by two processors. Eaton attempts NO SOLUTION to this problem, as Eaton clearly does NOT teach two processors able to access the SAME memory location at the same time. Eaton makes it clear that this cannot be done, as if it has the potential of occurring, Eaton teaches that one processor gets priority to exclusive access to the memory location, to the complete exclusion of the other. (See, e.g., Eaton, col. 4, lines 20-22)

Eaton teaches no need for the two processors to inter-communicate, and thus there is absolutely no motivation to combine a mailbox technique with Feemster. It is respectfully submitted that such combination is improper.

Even presuming that the combination of Eaton's separate memory stack technique with Feemster's mailbox between processors makes sense and is proper (which it is not), it STILL fails to teach the present invention.

For instance, at best, a combination of Eaton with Feemster would result in two processors having data stacks in common memory, AND mailboxes between one another. Even the combination of Eaton with Feemster would still fail to teach first and second mailbox portions **both** defined at least in part over **common** memory addresses, the first mailbox addressably filling upward through to a **highest** physical address of the common memory, and the second mailbox addressably filling downward through to a **lowest** physical address of the common memory, as claimed by claims 1-7. Moreover, the combination of Eaton with Feemster would still fails to teach or suggest placement of the mailbox in a **contiguous** block of **shared** memory, much less a first mailbox addressably filling upward through to a **highest** physical address of the **common** memory,

and a second mailbox addressably filling downward through to a **lowest** physical address of the **common** memory as claimed by claims 8-17.

For at least all the above reasons, claims 1-17 are patentable over the prior art of record. It is therefore respectfully requested that the rejection be withdrawn.

Conclusion

All objections and rejections having been addressed, it is respectfully submitted that the subject application is in condition for allowance and a Notice to that effect is earnestly solicited.

Respectfully submitted,

William H. Bollman
Reg. No. 36,457

MANELLI DENISON & SELTER PLLC
2000 M Street, NW 7TH Floor
Washington, DC 20036-3307
TEL. (202) 261-1020
FAX. (202) 887-0336